design ideas

Edited by Bill Travis and Anne Watson Swager

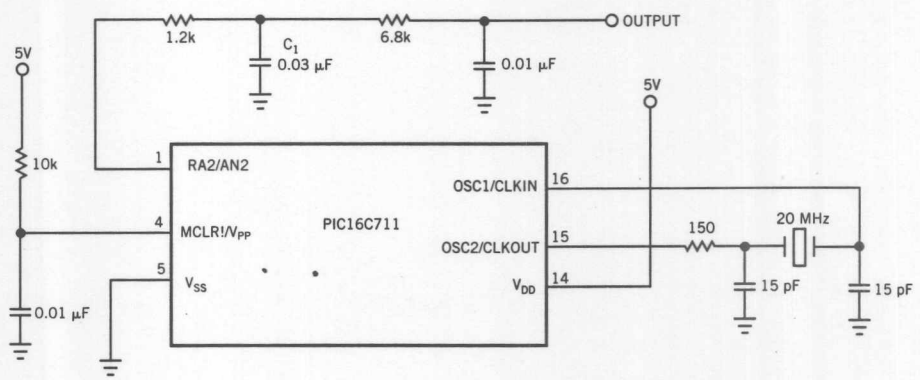
NCO technique helps μC produce clean analog signals

Steve Ploss, Veridian Corp, Wright Patterson AFB, OH

A RECENT DESIGN IDEA described a method for producing an analog voltage from one digital output of a μC ("Generate an analog signal with a μC," *EDN*, Oct 22, 1998, pg 108). The method involves generating a PWM output with a controlled duty cycle and filtering the switching waveform with a simple single-pole RC filter. Although this method provides an accurate dc output with 8 bits of resolution, it requires a filter with a low cutoff frequency to reduce the ripple to less than 1 LSB.

An alternative method doesn't have this problem. The circuit in **Figure 1** and the corresponding control program borrow a technique from direct digital synthesizers. The technique consists of a numerically controlled oscillator (NCO) that distributes the duty cycle as evenly as possible across the main period of the output, which is 256 clocks for both the NCO and PWM approaches. **Figure 2a** and **Figure 2b** illustrate the operation of the NCO and the PWM methods, respectively, with the duty cycle set to 10/256. **Figure 2a**'s NCO digital output has the same duty cycle as **Figure 2b**'s PWM output but distributes the duty cycle evenly across the period.

Figure 1



A numerically-controlled-oscillator (NCO) technique allows a μC and a handful of passive components to output analog signals that, after filtering, have a ripple amplitude that is constant over changes in duty cycle.

The benefit of the NCO is that the ripple amplitude after filtering is almost constant with changes in the duty cycle. In contrast, the PWM method has a ripple amplitude equal to that of the NCO approach at the lowest duty cycle, and the ripple worsens at midscale.

Figure 3 compares the expected output for each of the two methods using the same duty cycle as before, 10/256. A simple IIR filter that simulates a single-pole RC low-pass filter, performed the filtering. The time constant for this filter is 256 clocks. This **figure** shows that the NCO output has much lower ripple than the PWM output at this output duty cycle (**Figure 3a**). As the DAC value approaches midscale, which corresponds to a duty cycle of approximately 128/256, the PWM ripple gets progressively worse, but the NCO ripple improves by as much as a factor of 2 (**Figure 3b**).

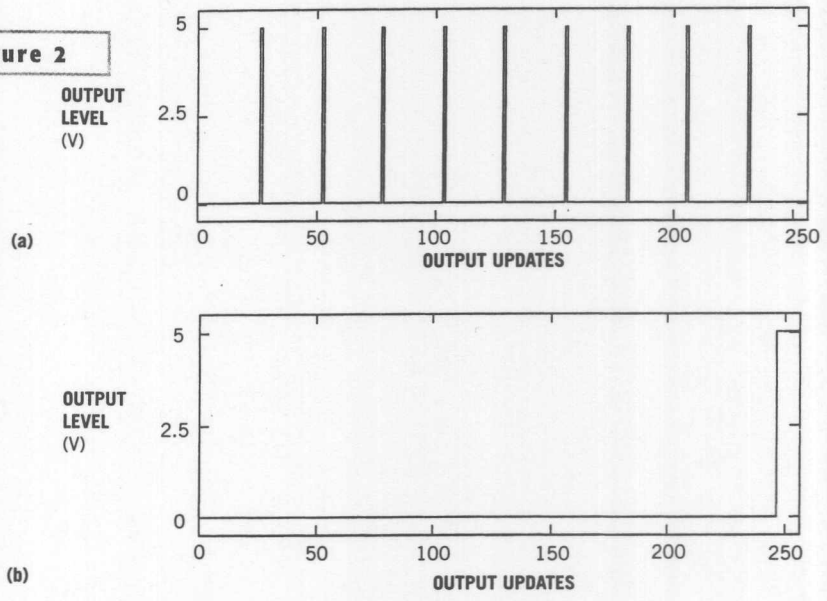
The accompanying listing to **Figure 1** runs on the PIC16C711, but, because the routine doesn't use the ADC and the TMR0 interrupts, you can use it with other processor types. (You can download the program from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2346.)

The DAC routine uses only two regis-

NCO technique helps μC produce clean analog signals **129**
μC generates musical sounds **132**
Continuity buzzer is frugal with power **134**
Stereo jack adds no-cost power/logic control **136**
S/H circuit minimizes aperture **140**

ters; the phase accumulator and the holding register form the DAC input value. Two additional registers provide sine-wave values to the DAC routine. On initialization, the phase accumulator sets to zero, and the DAC value sets to midscale (0x80). On each update, the value of the DAC register adds to the phase accumulator. When the phase accumulator rolls over—an event signaled by the setting of the carry bit—the circuit sets the output high for one update. The process then continues indefinitely. To generate a sine wave, the program counts the number of times it loops and every 32nd time it retrieves the next value from the look-up table. The table holds 16 values of a full cycle, so the period of the sine wave is 512 times the loop time, plus a small amount of time for the branch out of the loop. With a 20-MHz crystal, the loop time is approximately 3.5 μsec, and the sine-wave frequency is approximately 625 Hz.

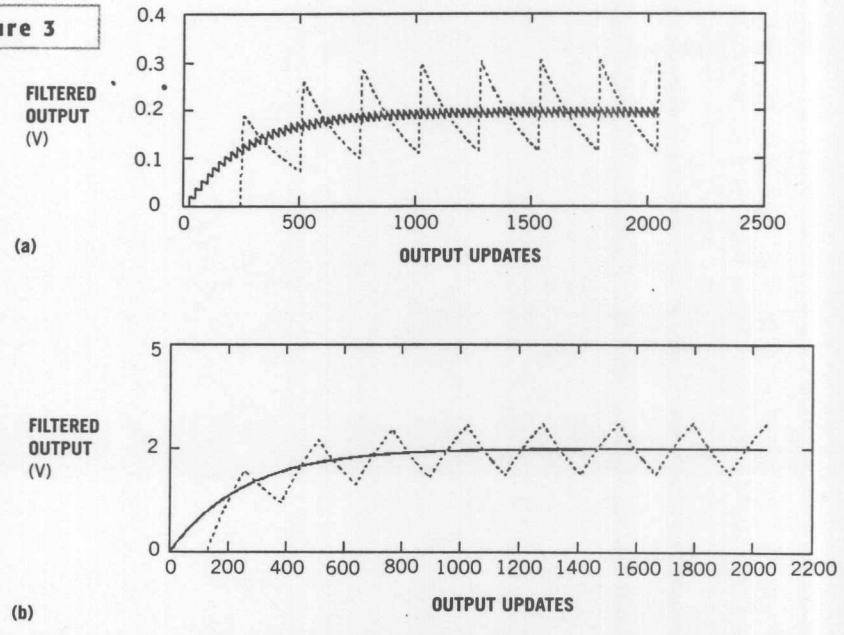
Figure 2



The NCO digital output (a) has the same duty cycle as the PWM output (b) but distributes the duty cycle evenly across the period.

To output an analog waveform, you need only to change the value of the DAC register. This change can happen at any time. The analog output almost immediately starts updating. When you use this method for your own application, remember to update the DAC output as often as possible to lower the ripple amplitude. The rate need not be precise, because you are merely setting a duty cycle—the period of which is relatively unimportant. More likely, you need to control the period of your analog waveform.

Figure 3



The output ripple that results from the numerically-controlled-oscillator (NCO) method—for example, at a duty cycle of 10/256—is much lower than that of the PWM method (a). As the DAC value approaches midscale—a duty cycle of approximately 128/256—the PWM ripple gets progressively worse, but the NCO ripple improves by as much as a factor of 2 (b).

If you can afford to use a μC with a timer interrupt, it's best to use that interrupt to determine when to change to the next DAC value. The rest of the time, you can continuously update the output. If you don't want to use the timer interrupt, you can update the DAC as part of a polling loop, as shown in this example, waiting for a loop counter to reach a predetermined value before changing the DAC value. However, you may want to make sure that the branch from the polling loop always takes the same amount of time, so that the DAC-value changes occur at a constant rate. Also, if the branch is going to take a comparatively long time, consider setting the output bit to a high-impedance state for the duration of the branch. If you do not take this precaution, the output will hold the last value—whether high or low—for the duration of the branch and could produce a transient on the filter output. (DI #2346)

To Vote For This Design, Circle No. 402